

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Aprendizaje Automático en los MOOCS

Rodrigo Barbero Martínez

Tutora Rosa María Carro Salas

Julio 2020

Aprendizaje Automático en los MOOCS

AUTOR: Rodrigo Barbero Martínez

TUTORA: Rosa María Carro Salas

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Julio de 2020

Resumen (Castellano)

En el contexto de la enseñanza no presencial, los MOOCs (cursos online abiertos masivos) suelen tener una enorme cantidad de estudiantes que, dependiendo de la temporalidad del MOOC, pueden realizar el curso en momentos muy diferentes, incluso en años distintos. En este tipo de cursos, los profesores no tienen el mismo tipo de *feedback* que en una clase presencial sobre el desarrollo de los mismos y suele ser más complicado detectar o prever las necesidades de cada estudiante. Existen foros donde los alumnos pueden realizar comentarios y preguntas, pero los mensajes que escriben pueden llegar a pasar desapercibidos dependiendo de la dinámica y estructura de cada MOOC. Por todo ello, sería recomendable introducir herramientas que puedan predecir automáticamente el estado de cada alumno con respecto al MOOC (si va bien encaminado o si existe riesgo de fracaso o de abandono) y avisar al profesor si se considera necesario, para que pueda intervenir si lo estima oportuno.

En este trabajo he estudiado algunas técnicas de procesamiento de lenguaje natural (Natural Language Processing o NLP) y de aprendizaje automático (Machine Learning o ML) para analizar, procesar y clasificar textos escritos. He seleccionado diversas técnicas con el objetivo de ver cuáles de ellas funcionan mejor para predecir el éxito, fracaso o abandono de los usuarios a partir de los textos que escriben. En particular, se han analizado los comentarios escritos por los estudiantes en los MOOCs de la UAM titulados “Quijote” y “Genética y la evolución”. He realizado diversas pruebas con las técnicas escogidas, afrontando los retos y desafíos que se describen en la memoria. He validado y analizado los resultados obtenidos y finalmente he llegado a la conclusión de que no se puede identificar a los tipos de usuarios de los MOOCs en base a los comentarios que escriben en ellos, al menos utilizando las técnicas y algoritmos presentados en este trabajo y los datos disponibles en los MOOCs analizados.

Abstract (English)

In the context of online classroom, MOOCs (massive open online courses) usually have a huge number of students who, depending on the timing of the MOOC, can take the course at very different times, even in different years. In this type of course, teachers do not have the same type of feedback as in a face-to-face class on their performance and it is usually more difficult to detect or anticipate the needs of each student. There are forums where students can make comments and questions, but the messages they write can go unnoticed depending on the dynamics and structure of each MOOC. For all these reasons, it would be advisable to introduce tools that can automatically predict the state of each student with respect to the MOOC (if it is on track or if there is a risk of failure or abandonment) and notify the teacher if it is considered necessary, so that they can intervene if deems it appropriate.

In this project I have studied some techniques of natural language processing (Natural Language Processing or NLP) and machine learning (Machine Learning or ML) to analyze, process and classify written texts. I have selected various techniques in order to see which of them work best to predict the success, failure or abandonment of users from the texts they write. In particular, the comments written by the students in the MOOCs of the UAM entitled

"Quijote" and "Genetics and evolution" have been analyzed. I have carried out various tests with the chosen techniques, facing the challenges that are described in the memory. I have validated and analyzed the results obtained and finally I have reached the conclusion that the types of users of MOOCs cannot be identified based on the comments they write, by using the techniques and algorithms shown in this work along with the data from these specific users and MOOCs.

Palabras clave (castellano)

Aprendizaje automático, MOOC, análisis de textos, comentarios de usuario, NLP, Python

Keywords (inglés)

Machine learning, MOOC, text processing, user comments, NLP, Python

Agradecimientos

En primer lugar, agradezco a Rosa mi tutora por guiarme en este proyecto.

A Diomedes, por ayudarme a llevar esta carga. Apoyándose cuando estaba triste y leyendo y corrigiendo este trabajo.

A Marina, por aguantarme y ayudarme a llevar mejor los problemas de la vida.

Y, por último, mi familia en general por todo el apoyo que me han dado.

Muchas gracias.

INDICE DE CONTENIDOS

1 INTRODUCCIÓN.....	9
1.1 Objetivo.....	9
1.2 Organización de la memoria.....	10
2 MARCO TEORICO	10
2.1 Massive Open Online Course	10
2.2 Natural Language Processing	10
2.3 Machine Learning.....	13
2.4 Modelos de Clasificación de Datos.....	14
2.5 Validar los Modelos.....	17
3 MARCO METODOLÓGICO	17
3.1 Herramientas.....	17
3.2 Esquema de Trabajo.....	18
4 ANÁLISIS DE RESULTADOS.....	21
4.1 Datos de Partida	21
4.2 Resultados.....	22
5 CONCLUSIÓN	37
REFERENCIAS	39
GLOSARIO	41
ANEXOS	I
Anexo A – Dataset Pv1 Comentarios	I
Anexo B – Dataset Pv1 Usuarios.....	II
Anexo C – Dataset Pv2 Comentarios.....	II
Anexo D – Dataset Pv2 Usuarios.....	III

INDICE DE TABLAS

TABLA 1: DATASET A NIVEL DE COMENTARIO	21
TABLA 2: DATASET A NIVEL DE USUARIO.....	21
TABLA 3: PRUEBAS PV1	23
TABLA 4: PV1 RESULTADO PRUEBA 1.....	24
TABLA 5: PV1 RESULTADO PRUEBA 2.....	25
TABLA 6: PV1 RESULTADO PRUEBA 3.....	26
TABLA 7: PV1 RESULTADO PRUEBA 4.....	27
TABLA 8: PV1 RESULTADO PRUEBA 5.....	28
TABLA 9: PV1 RESULTADO PRUEBA 6.....	29
TABLA 10: PV1 RESULTADO PRUEBA 7.....	30
TABLA 11: PV1 RESULTADO PRUEBA 8.....	31
TABLA 12: PRUEBAS PV 2.....	32
TABLA 13: PV2 RESULTADO PRUEBA 1.....	32
TABLA 14: PV2 RESULTADO PRUEBA 2.....	33
TABLA 15: PV2 RESULTADO PRUEBA 3.....	34
TABLA 16: PV2 RESULTADO PRUEBA 4.....	35

INDICE DE FIGURAS

GRÁFICO 1: DATASET A NIVEL DE COMENTARIO.....	22
GRÁFICO 2: DATASET A NIVEL DE USUARIO.....	22

1 INTRODUCCIÓN

A día de hoy muchas entidades, tanto públicas como privadas, dan soporte a la interacción con sus usuarios o clientes a través de sus plataformas o perfiles en redes sociales, que se utilizan diariamente. Dentro de dichas plataformas existe mucha información en forma de textos o comentarios, que puede permanecer oculta. Ejemplos de ello son: las opiniones políticas de los usuarios de Twitter, que realizan multitud de comentarios en dicha plataforma; las valoraciones que hacen los clientes de las compañías sobre sus productos o servicios; o las dudas, comentarios y opiniones que escriben los estudiantes en plataformas de aprendizaje online. Es por eso que muchas de estas entidades realizan estudios de los datos guardados en sus plataformas, siendo éstas un interesante medio para obtener información. En particular, el análisis de los textos escritos por los usuarios permite saber más de ellos, estudiar su perfil y mejorar las plataformas, servicios o recomendaciones ofrecidas, adaptándolos, en su caso, a las necesidades de cada uno de ellos.

Los *Massive Open Online Course* son cursos online donde los profesores no tienen un el mismo *feedback*, que en una clase presencial sobre el desarrollo de sus alumnos. Suelen tener una enorme cantidad de estudiantes que, dependiendo de la temporalidad del MOOC, pueden realizar el curso en momentos muy diferentes, incluso en años distintos. Existen foros donde los alumnos pueden realizar comentarios y preguntas, pero pueden llegar a pasar desapercibidos. Para ello sería recomendable introducir herramientas que pudieran predecir el estado de los alumnos y avisar al profesor si fuera necesario.

La utilidad de dichas herramientas es el manejo de los datos que se quieren procesar. Si se estuviera centrando en textos se utilizarían técnicas de *Natural Language Processing*, que permiten realizar un análisis de palabras y textos.

En el análisis de textos, si es necesario, se deben realizar modificaciones para eliminar términos o datos que se crean que van a generar problemas en el análisis.

Para realizar el aprendizaje de los datos ya procesados se usan técnicas de *Machine learning*, que obtienen conocimiento a partir de la información disponible y utilizan algoritmos de aprendizaje.

La idea es combinar estas técnicas y explorar cómo se obtendrían los mejores resultados al usar ambas aproximaciones, realizar pruebas iterando por las técnicas usadas y validar si los resultados son mínimamente exitosos, además de útiles en el contexto de los MOOCs

1.1 Objetivo

El objetivo del presente trabajo es la creación de un modelo de clasificación, que sepa predecir si un alumno que está realizando un MOOC va a aprobar o fracasar, tanto si es por suspender como por abandono, usando como fuente de conocimiento los comentarios que ha escrito en el MOOC.

1.2 Organización de la memoria

Este trabajo consta de cuatro secciones: En la primera se presenta el marco teórico, haciendo hincapié en las técnicas y tecnologías utilizadas en este trabajo. La segunda sección describe la metodología aplicada. En la tercera se desarrollarán las pruebas realizadas y los resultados obtenidos aplicando lo anteriormente escrito en el Análisis y por último, una conclusión final del trabajo.

2 MARCO TEORICO

En el siguiente apartado se explicará qué es un MOOC, realizando una descripción de las peculiaridades que le son propias, tales como contexto, objetivo y participantes, entre otras; cómo funciona el procesamiento del lenguaje natural (NLP); y algunas técnicas, algoritmos y herramientas de aprendizaje automático (ML) como vecinos próximos (KNN), regresión lineal, etc.

2.1 Massive Open Online Course

Un Massive Open Online Course (MOOC) es un curso online de libre acceso y masivo (Jordan, 2015), normalmente gratuito, que se puede encontrar en múltiples instituciones privadas y públicas (p ej. la Universidad Autónoma de Madrid). De una gran diversidad temática, habitualmente están compuestos por recursos digitales (vídeos, imágenes, textos), actividades que permiten practicar lo aprendido y exámenes miden los conocimientos adquiridos por el alumnado. En las plataformas que dan soporte a los MOOCs se reconocen dos tipos de usuarios: Los profesores que administran los cursos y los alumnos que realizan las actividades. Este último grupo se subdivide a su vez en dos partes: Los primeros son los usuarios que llegan al final del curso, que se etiquetaron como “finalizadores” y las personas que abandonan el MOOC, a quienes a partir de ahora se llamará “abandonadores”.

Durante el periodo de duración del curso, se almacena toda la información sobre las interacciones del usuario en la plataforma, que pueden ser:

- Interacciones con los videos.
- Interacciones en los exámenes.
- Comentarios en los foros.
- Calificaciones relacionadas con el curso.

La información en formato texto disponible para ser analizada son los comentarios en los foros. A continuación, se explicará una posibilidad para el análisis de texto basada en el lenguaje y su estructura.

2.2 Natural Language Processing

Para el análisis de textos a nivel de lenguaje se utiliza Natural Language Processing (NLP). Es una rama de la ciencia de la computación, encargada de la traducción de las palabras y textos en expresiones que una máquina pueda analizar sin ayuda humana. (Alias y Cassanelli, 2019)

Esta herramienta es necesaria, para que el un autómatas pueda comprender el funcionamiento del lenguaje humano, dado que este presenta un nivel muy elevado de abstracción.

Este enfoque presenta un problema, ya que en la transformación de los textos se puede llegar a perder significado. Esto es debido a un desconocimiento del contexto o por el uso de expresiones muy complicadas de traducir, generando una mala traducción de los términos.

Para realizar esta tarea existen múltiples técnicas. Los más utilizados y que se realizaran en este trabajo son los siguientes:

- Bag of Words
- Term Frequency–Inverse Document Frequency (*TFIDF*)
- Análisis de sentimientos
- Doc2vec

2.2.1 Bag of Words

Esta técnica funciona creando un vector usando una o varias palabras (n-gramas)¹, calculando la frecuencia de su aparición en los textos (Hanna M, 2006). Esta técnica necesita de modificaciones de las palabras antes de hacer un análisis. Las alteraciones son producidas debido a:

Primeramente, las palabras con una alta frecuencia de aparición en el texto, pero carentes de relevancia para la semántica de las frases. Estas palabras son conectores, artículos... Estos términos son denominados *stopwords*. Para resolverse este problema, se eliminará sus apariciones en el texto, para que no se tengan en cuenta en el vector.

Otro de los motivos, es la gran cantidad de palabras que pueden aparecer en el vector. Esto puede producir problemas de almacenamiento y procesamiento de los datos finales tras la transformación. Para ello se debe realizar una agrupación de palabras, usando el lexema como punto de unión. De esta manera se reduce la cantidad de términos a almacenar y a procesar

Con estos cambios, *bag of word* podría seguir teniendo problemas de almacenamiento y alto procesamiento generado por la cantidad de palabras encontradas. Por otro lado, cuantos más textos se procesen, más términos se van a encontrar. Por tanto, se debería reducir utilizando otros filtrados.

2.2.2 Term Frequency–Inverse Document Frequency (TFIDF)

El funcionamiento de TFIDF está basado en la búsqueda de frecuencias de palabras o conjuntos de palabras en un texto (Soucy y Mineau, 2015). De forma paralela es necesario eliminar los términos denominados *stopwords* del análisis.

¹ Agrupación de n palabras.

A diferencia de lo que ocurriría con el método anterior, con esta técnica no se elegirán todas las palabras, sino que se hará un filtrado. Para llevar a cabo este proceso se elegirán los términos que sean menos frecuentes dentro de los textos. Pero que su aparición en los comentarios sea más frecuente. Con esto se obtendrían las palabras que aportan más significado dentro del contexto donde se han escrito los textos.

La principal ventaja de TFIDF frente a Bag Of Words es la mejor optimización del análisis, en cuestiones de almacenamiento y procesamiento de información. Porque en TFIDF se decide la cantidad de palabras que se van a recoger según su ranking de frecuencia.

No obstante, aunque identificamos bastante bien a las palabras que van a tener mayor importancia en el contexto de los textos, se podrían estar omitiendo términos por una configuración inadecuada. Como se está realizando un filtrado según su frecuencia, podría haber palabras que, aunque apareciendo en menos ocasiones, podrían representar mejor al texto, pero no llegan a aparecer en la expresión.

2.2.3 Sentimientos

Esta técnica tiene como objetivo la traducción de los sentimientos plasmados en palabras y conjuntos de palabras a conceptos entendibles para una máquina (Baviera, 2017).

Para realizar esta tarea se necesitan dos herramientas. La primera son diccionarios de sentimientos, que asocian a las palabras a un valor que refleja la polaridad del sentimiento. Generalmente este valor oscila entre -1 (negativo) y 1 (positivo), donde 0 indica neutralidad. La segunda herramienta da soporte al análisis sintáctico de la expresión, utilizando redes neuronales. Juntándose estas dos aproximaciones, en principio se podría identificar la polaridad del sentimiento transmitido por cualquier conjunto de términos.

Existen muchas aplicaciones y librerías para dar soporte a esta tarea. Principalmente se basan en diccionarios creados a mano que identifican las palabras con un valor de la polaridad del sentimiento y en una implementación de una red neuronal.

El problema que presenta este acercamiento es la necesidad de diccionarios de palabras del lenguaje que se quiere analizar. Si no existe una librería compatible que se pueda usar para identificar el sentimiento de una palabra, se debería generar un diccionario propio o extender alguno de los existentes, analizando las palabras posibles, asignándoles un valor que indique la polaridad del sentimiento e incorporándolas al diccionario correspondiente.

2.2.4 Doc2vec

Esta técnica se basa en el uso de la herramienta de Google Doc2vec. Esta lo que realiza es, a través de redes neuronales, una predicción del siguiente texto que va a generar el usuario según sus escritos anteriores, creando así una estructura que identifica como escribe el usuario. Esta es una matriz $N \times 5$, donde N es el número que varía. (Han Lau y Baldwin, 2016)

De esta manera Doc2vec se podría utilizar para analizar varios textos de una misma persona y descubrir cómo escribe un usuario.

Los problemas que presenta esta técnica es el desconocimiento total de su funcionamiento interno. Además, la N anteriormente mencionada varía según el algoritmo lo crea necesario. Por tanto, si se quiere hacer una comparativa entre usuarios se deberá hacer modificaciones a dicha matriz.

Tras esta exploración de los diferentes métodos de análisis de textos, se procederá a definir qué es Machine Learning y cuáles son sus principales herramientas.

2.3 Machine Learning

Machine Learning (ML) puede definirse como campo de la computación encargada de dotar de la capacidad a autómatas para obtener conocimiento o habilidades a través de técnicas de aprendizaje. Estas se fundamentan en conjuntos de datos anteriormente formateados, para el buen entendimiento del autómata. A esta entidad se la nombrará un modelo de aprendizaje automático.

Esta tarea se realiza de forma automática, es decir sin que el ser humano intervenga en el aprendizaje. Es por ello, que el modelo debe ser capaz de adaptarse a su base de conocimiento y aprender de ella.

Por tanto, como se describe en (Moreno, Antonio., Armengol, Béjar Javier, Luís, Gavalda, Gimeno, López, Martín y Sánchez 1994). el aprendizaje se podría definir como “*Aprendizaje=selección + adaptación*”, donde selección se refiere al apartado del aprendizaje usando la base de conocimiento y adaptación se refiere a la capacidad de cambio ante nuevos datos.

Existen múltiples técnicas de aprendizaje automático. Dado el objetivo del presente trabajo se describirán las que se podrían usar en ese contexto:

- Aprendizaje analítico
- Aprendizaje inductivo
- Conexionismo

2.3.1 Aprendizaje Analítico

Esta técnica se basa en la obtención de conocimiento analizando una base de conocimiento. De esta manera el algoritmo debe aprender cada uno de los casos para entender por qué se producen y dan el resultado mostrado. Con esto el algoritmo podría obtener información del contexto en el que se le está enseñando.

2.3.2 Aprendizaje Inductivo

Esta técnica se basa en inducir de un dato final las características relacionadas con esa información. En este tipo de aprendizaje se desconocen en su totalidad los atributos de estos datos.

2.3.3 Aprendizaje Conexionismo

Es una red de nodos, donde cada conexión entre nodo tiene un peso. La idea es que, por cada dato aprendido, en base al resultado del aprendizaje (si el autómata identificó de forma correcta la información) se realicen unas modificaciones a los pesos anteriormente mencionados, intentando corregir si hubo un fallo o reforzar el dato aprendido.

Una vez explicadas las técnicas presentes en el análisis de textos, se hablará de las tres formas de utilizarlas:

- Aprendizaje por refuerzo
- Aprendizaje supervisado
- Aprendizaje no supervisado

2.3.4 Aprendizaje por Refuerzo

En el aprendizaje por refuerzo, se utilizan técnicas basadas en premiar y castigar para ayudar a la máquina a discernir entre aciertos y fallos. De esta forma se fomenta la búsqueda de un alto nivel de recompensa por parte del programa.

2.3.5 Aprendizaje Supervisado

La idea es dirigir el aprendizaje para la creación del modelo usando un conjunto de datos recogidos anteriormente y clasificados. De esta manera el modelo iría aprendiendo por cada dato cuya clasificación conoce, y sus predicciones serían calculadas conforme la clasificación conocida.

2.3.6 Aprendizaje No Supervisado

Este tipo de aprendizaje se basa en el descubrimiento de nueva información sobre los datos que conoce el modelo de clasificación. Al contrario que en el caso anterior no hay evaluación posible y se trata de remarcar los datos o atributos que les parezca a los autómatas como relevantes.

En el presente trabajo se ha utilizado aprendizaje supervisado, ya que los datos de los MOOCs ofrecían toda la información necesaria y existen muchos modelos de aprendizaje que se adaptan mejor a esta idea. Existen varios métodos para la clasificación de los datos, que se resumen continuación.

2.4 Modelos de Clasificación de Datos

Los modelos de clasificación se obtienen utilizando algoritmos de aprendizaje automático orientados a la clasificación de datos en basándose en etiquetas o clases. Dependiendo del tipo, se tiene un acercamiento distinto al procesamiento de los datos y, además, hace falta configurar cada uno de ellos de forma óptima para el contexto en el que se encuentra el autómata. (Jurafsky y Martin, 2019)

- Vecinos próximos (KNN)
- Redes neuronales (NN)
- Árboles de decisión (DTC)
- Random Forest(RFC)

- Regresión lineal(LR)
- *Support Vector Machine (SVM)*

2.4.1 Vecinos Próximos (KNN)

Este modelo se encarga de clasificar un dato basándose en la distancia entre éste y la información disponible en su base de conocimiento. Tras esto se seleccionan los K más cercanos y se decide la clase de ese dato observando cuál es la clase mayoritaria en los datos seleccionados. Como siempre hace falta que haya una clase con mayor número de casos, K debe ser impar.

La separación entre los datos desconocidos y conocidos, se podrá determinar calculando la distancia Euclídea o la Manhattan. En ambas se usan el teorema de Pitágoras para el cálculo, pero el primero se basa en la hipotenusa y el segundo en los catetos.

Cuanto más grande sea K más sensible será el clasificador., porque se estaría teniendo en cuenta en la clasificación un valor elevado de casos de la base de conocimiento. En cambio, si el valor de K es muy bajo, apenas habría datos con los que comparar dentro del *dataset*.

Una de las ventajas de este modelo es que no tiene que obtener información adicional de los datos. Pero a la hora de clasificar tiene que usar toda su base de conocimiento para poder medir la distancia. Además, el valor K no es fácil de determinar, siendo necesario hacer pruebas para hallar el K óptimo.

Por tanto, existe una dificultad a la hora de almacenar la base de conocimiento y de procesarla, ya que se necesita mantener todos los datos conocidos. Y, además, cada vez que se quiera aprender es necesario recorrer todos ellos para saber cuáles son los vecinos próximos.

2.4.2 Árboles de Decisión

Son árboles binarios creados a partir de una inducción de los datos, mediante la cual, se van generando reglas que hacen un filtrado respecto a los atributos de los datos. Estos separadores son los nodos del árbol y los nodos hoja, son los resultados de la clasificación.

Un ejemplo es, para el atributo de edad los que son menores de 90 años son de clase 0 y los mayores de clase 1. De manera iterativa se va generando todo el árbol, donde sus nodos hojas serían la clase que se clasificaría si se llegase ahí.

Para configurar este clasificador hay que determinar la profundidad que puede alcanzar como máximo. Cuanto más profundo sea el árbol, será mucho más sensible y cuanto menos profundo más robusto, pudiendo generalizar mejor ante casos no conocidos. A diferencia del KNN aquí sí que hay un aprendizaje previo a la clasificación, ya que se debe generar el árbol y guardarlo. Pero no es necesario realizar esta operación cada vez que se quiera clasificar, a no ser que haya información adicional a tener en cuenta en el aprendizaje.

2.4.3 Random Forest

Este modelo se basa en la creación de N árboles de decisión, que aprenden a partir de la base de conocimiento. A la hora de clasificar un dato, todos los árboles anteriormente mencionados realizan una clasificación. La clase mayoritaria es el resultado del aprendizaje de Random Forest. Se reconoce como uno de los mejores clasificadores por sus buenos resultados (Wainberg, Alipanahi y Frey, 2016)

2.4.4 Regresión Lineal

El objetivo de la regresión lineal es la separación de los modelos por planos, para poder diferenciar las clases existentes. Según vaya aprendiendo la entidad irá modificando dichas estructuras geométricas para poder separar de la mejor manera todas las clases.

Se puede configurar con la constante de aprendizaje, cuanto más grande sea los planos oscilarán más y serán muy sensibles a los nuevos modelos, en cambio, si es muy baja, apenas será robusto.

2.4.5 Support Vector Machine SVM

La idea es generar una serie de vectores que forman hiperplanos que separen la muestra en tantas clases como haya.

Para realizar las predicciones se observará en qué intervalo entre estos hiperplanos está el punto a clasificar. Para aprender, se debe especificar qué formas pueden obtener los hiperplanos (linear, rbf y polinomial) y su capacidad de aprendizaje. Como ocurre en los casos anteriores, cuanto mayor sea el modificador de los hiperplanos, mayor más oscilarán las predicciones y cuanto menor sea, será menos sensible.

Por tanto, lo que busca este clasificador es recoger los valores en márgenes entre hiperplanos, intentando maximizar las clases con las que ha aprendido en los espacios que se clasificaría.

2.4.6 Red Neuronal

El objetivo de los modelos basados en redes neuronales es crear una red de nodos basándose en el aprendizaje con conexiones. Cada nodo se llama neurona y éstas se agrupan en capas ocultas dentro de un autómata. Para la creación del programa, se debe configurar el número de neuronas por capa y el número de capas.

Las neuronas tienen poca capacidad de procesamiento y se unen con otras con un peso asociado. Cuando se reciben los datos de entrada, se genera una respuesta en la que influyen los valores asociados a las conexiones y para aprender de la respuesta se realiza un “rollback²” modificando los pesos conforme al acierto o fracaso de la predicción.

² Realizar ingeniería inversa, es decir, desde el resultado final llegar al principio.

La capacidad de adaptación se debe marcar con un valor de aprendizaje, como en los casos anteriores. Cuanto mayor sea, más oscilará, siendo más sensible y cuanto menor sea, será más robusta.

2.5 Validar los Modelos

Para comprobar la efectividad de los modelos habrá que dividir la muestra de datos en dos secciones: la primera la encargada de entrenar el modelo elegido y la segunda para poder comprobar la efectividad de este.

Para romper la aleatoriedad sobre qué valores se utilizan en cada sección, se usa una técnica llamada *Cross validation*, basada en la separación de los datos accesibles en tantos grupos como se quiera. Después se va rotando qué sección va a ser la que pruebe la efectividad, y el resto servirá para entrenar el modelo. Así, la media de todas las pruebas será la respuesta a cómo se ha comportado con todos los datos en global.

Habiéndose explicado cómo funcionan las técnicas y algoritmos que se utilizan para el análisis y el aprendizaje, se procederá a la explicación sobre la metodología utilizada en este trabajo.

3 MARCO METODOLÓGICO

En este apartado se describen las herramientas y metodologías seleccionadas para el presente trabajo, así como sus principales utilidades y puntos de relevancia. Para ello se subdividirá en los siguientes apartados:

- Herramientas de trabajo
- Esquema de trabajo

3.1 Herramientas

Para el presente trabajo, se utilizaron varias herramientas tecnológicas relacionadas con el procesamiento del lenguaje natural, el aprendizaje automático y la programación de soluciones concretas para ambos.

3.1.1 Python

Es un lenguaje de programación orientado a objetos, cada vez de mayor relevancia debido a su sencillez de programación, la variedad de librerías y herramientas que se han ido creando alrededor del lenguaje y que se encuentran disponibles, y su potencia.

Dentro de este entorno la librería más relevante es “Scikit-Learn” (Pedregosa, Varoquaux, Gramfort Alexandre, Thirio, 2011). Esta proporciona una variedad elevada de utilidades para trabajar con *Machine learning* y se puede usar para implementar los clasificadores anteriormente mencionados.

Además, existe otra librería que soporta este lenguaje de programación llamada “*TextBlob*” (Loria, 2020), que permite la administración y análisis de texto necesarios para traducir el lenguaje de los comentarios de los MOOCs a expresiones entendibles para un programa.

A su vez, para administrar la conexión entre ambas partes se ha utilizado unas librerías de manejo de *dataset* llamada *Numpy* que da soporte a el almacenamiento de información y administración de una manera sencilla.

3.1.2 Visual Studio Code

Visual Studio Code es un entorno de programación, que tiene una gran afinidad con una alta variedad de lenguajes de programación. En particular, con el lenguaje elegido aporta atajos y ayudas a la hora de programar. Además, proporciona una herramienta de depuración para Python.

3.2 Esquema de Trabajo

A continuación, se describe el acercamiento metodológico que se realizó durante la elaboración del presente trabajo. Para ello se subdividirá en dos bloques: Criba de datos y aprendizaje de los clasificadores.

3.2.1 Criba De Datos

Para realizar esta tarea se tuvieron que agrupar todos los comentarios de los usuarios de los MOOCs proporcionados para este trabajo. Una vez hecha esta tarea, se generaron dos *dataset* sin procesar a dos niveles, usando distintas unidades en el *dataset*: Utilizando los comentarios y tomando los usuarios como unidad mínima.

Durante esta fase además se etiquetó cada dato obtenido con una clase dependiendo de su estado al finalizar el curso. Se distinguen 3 estados:

- Aprobado
- Suspenso
- Abandono

Con esto se plantearon dos contextos distintos: El primero es que el suspenso y el abandono se consideraban el mismo tipo de fracaso y se etiquetaron ambos casos como **clase 1** y aprobados como **clase 0**.

En el segundo, se supuso que el suspenso y el abandono eran maneras distintas de fracasar en el MOOC, quedando el primer tipo como clase 1 y el segundo como **clase 2**.

Una vez agrupados los datos, se realizó la traducción del *dataset* usando todas las técnicas de NLP anteriormente descritas. Con la salvedad que solo doc2vec trabajó a nivel de usuario y el resto se centró a nivel de comentario.

Para las técnicas de TFIDF y de sentimientos se necesita realizar una configuración previa al análisis. Se realizaron pruebas con diversas configuraciones, hasta encontrar que los mejores resultados se obtenían con la siguiente configuración:

- TFIDF: 100 palabras
- Sentimientos: Se generaron 4 atributos para el vector: Calculando el sentimiento medio de todas las palabras y frases de los comentarios y el global del texto, además de considerar la cantidad de frases.

Durante el procesamiento surgieron los siguientes problemas: Bag of words generó vectores de gran tamaño: Como se explicó anteriormente, *bag of words* tiene un problema de almacenamiento debido a la gran cantidad de términos que se pueden guardar de un texto. Se realizaron las modificaciones mínimas para utilizar este método y se obtuvieron vectores que excedían la capacidad de memoria que el sistema operativo cedía a Python.

Es por ello que se realizó un filtrado en referencia al contexto de este trabajo. Este se basó en eliminar aquellos términos cuyo significado fuera neutro. El tamaño del vector se redujo un 97,7%.

El problema de esta solución fue que esos términos encontrados no tenían por qué tener una alta frecuencia en los textos. Esto produjo una gran cantidad de comentarios cuyo vector asociado estaba completamente formada por ceros y, por ello, se perdió una gran cantidad de datos. Por esto se decidió omitir este método de NLP para las pruebas finales.

Gran cantidad de abandonadores: Durante las pruebas se observó que el número de usuarios que abandonaban el curso era muchísimo mayor que los finalizadores del mismo. Este desbalanceo generó que la clase 0 estuviera en desventaja en los dos contextos explorados, tanto si los datos se dividían en 2 clases como si se dividían en 3.

Para resolver este problema se pensó en balancear el *dataset* utilizando la clonación de las clases minoritarias. Esto consiste en replicar datos existentes dentro del *dataset* para que, a la hora de realizar el aprendizaje de los datos, se tenga más en cuenta a las clases minoritarias, antes en desventaja.

Además, también se pensó en observar otros puntos de vista modificando los datos, surgiendo las siguientes alteraciones:

Eliminar abandonadores: Este caso será cuando solo se quiera observar los resultados bajo el punto de vista de los usuarios que llegaron al final del MOOC. Se pensó de esta manera porque los comentarios que pudieran ofrecer las personas que abandonan podrían introducir una información no útil. Se desconoce el porqué del abandono de los usuarios. Incluso puede no deberse a un problema con la temática del MOOC y por ello los comentarios asociados serán muy variados, generando una alta incertidumbre.

Con este cambio, se preservarían los datos que se pudieran clasificar de manera sencilla mejorando el aprendizaje. Además, se eliminaría ruido en la base de conocimiento de los autómatas.

Eliminar usuarios con solo un comentario: Existen MOOCS que obligan a sus usuarios a realizar un comentario en su curso como primera actividad. Además de eso, las personas que solo realizan un comentario y luego abandonan el curso pueden introducir ruido y por ello, afectar a la hora de realizar el aprendizaje.

Todas estas modificaciones y distintos puntos de vista no fueron únicos, sino que se realizaron mezcla para observar su comportamiento en el futuro aprendizaje. Que se comentará a continuación.

3.2.2 Aprendizaje y Prueba

Para realizar la prueba de todos los posibles clasificadores que se han mencionado en la sección 2.4 (Modelos de clasificación de datos), se creó un script. Para ello primero se tuvo que comprobar cuál era la mejor configuración para cada clasificador, con los datos obtenidos. Después se pudo probar usando la técnica de cross-validation con 3 grupos de datos, generando los siguientes valores a analizar:

- Precisión
- Recall
- Sensibilidad
- Matriz de confusión

Para realizar la configuración de los clasificadores, se realizaron pruebas masivas para comprobar en base a los datos de partida, qué configuración era la mejor. Para ello se procesó la información a nivel de comentario y de usuario.

Para tomar estos valores como una buena configuración, se supuso que, si en la totalidad de los datos esta configuración era correcta, tras las modificaciones que mejoraban ese aprendizaje, el valor seguiría siendo correcto en ese contexto.

Con todo esto en mente, se realizó una reflexión sobre el coste del fallo del aprendizaje de los autómatas y si el resultado se obtuvo en un entorno balanceado. Esto se muestra en la matriz de confusión de cada una de las pruebas.

Esta reflexión tendrá como objetivo buscar un modelo de clasificación que identifique de forma correcta a los usuarios que suspendieron o abandonaron los cursos. Se decidió poner el foco en identificar bien a estos usuarios, porque el coste que supone atender innecesariamente a un alumno que no tiene problemas es más bajo y tiene menor repercusión que no atender a un alumno que sí los tenga. Por tanto, se buscarán los siguientes patrones, en orden de importancia:

- Muestra balanceada.
- Recall de la clase fracaso sea superior al 50%.
- Sensibilidad de la clase fracaso superior al 50%.
- Precisión supere 50%.
- Diferencia de recall menor del 30%

Estos valores se eligieron así, porque como se ha comentado anteriormente, resulta más grave clasificar de forma incorrecta a una persona que va a suspender (como si fuera a aprobar) que equivocarse en el sentido contrario. Aun así, lo que se buscó no es un modelo de clasificación que siempre decida la clase fracaso. Para eso se comprobaron los recalls resultantes y se decidió que superar el 30 % distanciaba demasiado la elección de ambas clases.

Durante las pruebas se observó un problema relacionado con el NLP para detectar sentimientos. Se detectó que la cantidad de palabras, frases y textos cuyo valor de sentimiento fuera distinto de 0 o cercano a este era muy baja. Eso quiere decir, que los usuarios que escribían en los foros no utilizaban expresiones que reflejan un alto nivel de sentimiento, tanto positivo como negativo o que la librería usada no era capaz de detectarlas.

Para solucionar este problema, se decidió juntar los sentimientos encontrados en todos los contextos posibles en un texto, es decir, a nivel de palabra, frase y comentario. Además, intentando descubrir nuevos atributos que pudieran ayudar en el aprendizaje, se introdujo uno nuevo: el número de frases presentes en el texto. Con esta nueva aproximación se observó una mejoría en los resultados finales.

4 ANÁLISIS DE RESULTADOS

4.1 Datos de Partida

El trabajo se inició con dos conjuntos de datos proporcionados, de forma anonimizada, por la oficina UAMx de la Universidad Autónoma de Madrid, correspondiente a los MOOCs del Quijote (Quijote501x) y de la Genética y la evolución (GenEvlx). Se recogió y unió la información obtenida de ambos y se etiquetaron conforme a 3 clases: en la clase 0 los usuarios que habían aprobado el curso, en la clase 1 los que habían suspendido y en la clase 2 los que habían fracasado. Además, se generaron dos *dataset* dependiendo del punto de vista (comentarios o usuarios).

Clase	Nº Comentario
0	461
1	353
2	811

Tabla 1: Dataset a nivel de comentario

Clase	Nº Usuarios
0	116
1	139
2	255

Tabla 2: Dataset a nivel de usuario

Como se puede observar en el gráfico 1 y en el gráfico 2, hay muchos más abandonadores (clase 2) que finalizadores (clase 0 y clase 1). Dentro de este último grupo hay más aprobados que suspensos. Pero la diferencia es menor comparada con los usuarios que finalizan y abandonan. Esto indica que el *dataset* está desbalanceado y si los datos no presentan patrones claros, se obtendría una baja precisión en las pruebas.

Se realizaron pruebas masivas para buscar en base a los datos de partida la configuración correcta, con los resultados mostrados en el esquema 1.

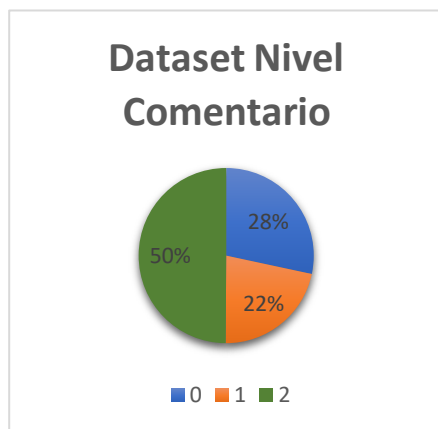


Gráfico 1: Dataset a nivel de comentario

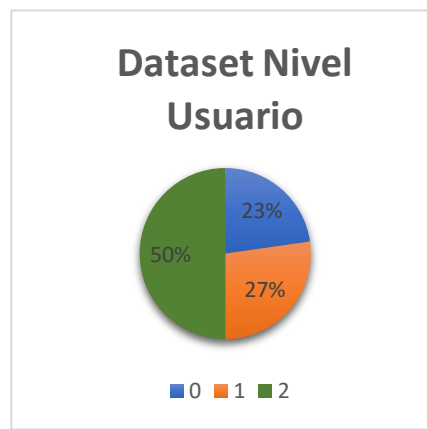



Gráfico 2: Dataset a nivel de usuario

Esquema 1: Resultados de la configuración

Clasificador	Configuración
KNN	<ul style="list-style-type: none"> • $K = 5$ • Distancia se calcula de forma Euclídea
NN	<ul style="list-style-type: none"> • Alpha 0.25 • 60 capas de 5 neuronas
LR	No se modificó la configuración de scikit-learn
DTC	No se modificó la configuración de scikit-learn
RFC	<ul style="list-style-type: none"> • Max-depth 140 • 300 árboles
SVC	Forma lineal

4.2 Resultados

Los resultados se presentan bajo dos puntos de vista: El primero, uniendo los datos de la clase 1 y clase 2, considerándose ambos como fracaso. Se referirá a este conjunto de pruebas como PV1. En el segundo punto de vista PV2 se consideran las tres clases por separado.

Para PV1 se organizaron varias pruebas, realizando distintas modificaciones en el *dataset*. El nombre de cada prueba y qué alteraciones se realizaron se muestra en la tabla 3 en la que el símbolo  es indicativo de modificaciones realizadas en el *dataset*:

Eliminar abandonadores: Se eliminan de la base de datos todos los usuarios que abandonaron el MOOC.

Balancear: Se generan copias de la clase minoritaria según su proporción, dado que varía según otras modificaciones.

Comentarios > 1: Solo se seleccionan los usuarios que han escrito más de un comentario en el MOOC.

Pruebas	ELIMINAR ABANDONADORES	BALANCEAR	COMENTARIOS > 1
p1	✗	✗	✗
p2	✗	✗	✓
p3	✗	✓	✗
p4	✗	✓	✓
p5	✓	✗	✗
p6	✓	✗	✓
p7	✓	✓	✗
p8	✓	✓	✓

Tabla 3: Pruebas PV1

A partir de ahora se presentan los resultados obtenidos. Cada uno de los casos será referenciado utilizando su nombre en la Tabla 3 (p1, p2, ...). Se hará un breve resumen de las modificaciones hechas en él. Se mostrarán los resultados por cada técnica de NLP empleada (Sentimientos, TFIDF y Doc2Vec), junto con los clasificadores empleados (KNN, NN, SVC, DTC RFC, LR). Por último, se incluye una breve conclusión global de los resultados. La información de la situación de las clases en cada prueba, se podrá encontrar en los anexos A y B.

Los resultados se mostrarán de la siguiente forma:

Técnica NLP: Técnica usada en esa prueba.

Recall 0: Resultado recall medio de la clase 0.

Recall 1: Resultado recall medio de la clase 1.

Sensibilidad 0: Resultado sensibilidad media de la clase 0.

Sensibilidad 1: Resultado sensibilidad media de la clase 1.

Precisión: Precisión media de las pruebas.

Error: Error medio de las pruebas.

Se recuerda que de las técnicas de NLP, solo trabajo a nivel de usuario Doc2vec, dado que su potencia se basa en el análisis de más de un texto. En cambio, el resto de métodos se basan en los comentarios en particular.

4.2.1 PV1

A continuación, se muestran los resultados de todas las pruebas realizadas en las que se abordó desde el punto de vista donde clase 2 se une a la clase 1 (PV1)

4.2.1.1 P1

En la tabla 4 se muestran los resultados de la p1, donde no se realizaron modificaciones a los datos disponibles.

Tabla 4: PV1 Resultado Prueba 1

TFIDF-100	Recall 0	Recall 1	Sensibilidad 0	Sensibilidad 1	Precisión	Error
KNN	3,48	75,17	21	71	50,75	49,25
NN	0	89,44	0	71,63	61,08	38,92
SVC	0	89,45	0	71,63	61,08	38,92
DTC	0	71,12	0	71,43	53,33	46,67
RFC	0	71,49	0	70,52	44,22	55,78
LR	0,3	90,19	0	70,52	44,22	55,78
Doc2Vec	Recall 0	Recall 1	Sensibilidad 0	Sensibilidad 1	Precisión	Error
KNN	0	85,32	0	77,09	62,85	37,15
NN	0	84,61	0	77,25	61,87	38,13
SVC	0	89,73	0	77,25	66,98	33,02
DTC	0	76,28	0	76,57	54,6	45,4
RFC	0	76,29	0	76,57	54,6	45,4
LR	0	89,52	0	77,25	66,78	33,22
Sentimiento	Recall 0	Recall 1	Sensibilidad 0	Sensibilidad 1	Precisión	Error
KNN	5,37	76,02	4,5	68,13	47,64	52,36
NN	2,15	82,97	5,12	68,67	52,81	47,19
SVC	0	94,37	0	68,81	63,19	36,81
DTC	0	66,48	0	65,73	37,93	62,04
RFC	0	66,48	0	65,73	37,93	62,06
LR	0	97,67	0	68,68	66,65	33,34

4.2.1.1.1 Conclusión:

Los valores *Recall* de la clase 0 de todas las pruebas presentan un bajo porcentaje, a diferencia de la clase 1, en la que se observan valores superiores al 66%. Esto genera una precisión superior al 34% en todas las pruebas. Se deduce de estos resultados que los clasificadores han optado por la clase dominante, la clase 1, a causa de que los datos presentan vectores muy parecidos en ambos bandos, generando confusión a la hora de aprender de ellos y decayendo en una búsqueda de la clase mayoritaria.

Por ello no se aceptó ningún clasificador como válido.

4.2.1.2 P2

En la Tabla 5 se muestran los resultados de la p2, donde se eliminaron los usuarios que solo habían escrito un comentario.

Tabla 5: PV1 Resultado Prueba 2

TFIDF-100	Recall 0	Recall 1	Sensibilidad	Sensibilidad	Precisión	Error
KNN	5,68	69,42	28,91	67,5	43,85	56,15
NN	0	67,54	0	66,48	37,75	62,25
SVC	0	74,18	0	67,56	43,7	56,3
DTC	0	67,25	0	66,5	37,45	62,55
RFC	0	67,69	0	66,47	37,9	62,1
LR	1,17	83,69	0	66,48	52,9	47,1
Doc2Vec	Recall 0	Recall 1	Sensibilidad	Sensibilidad	Precisión	Error
KNN	1	69,98	1,81	63,91	39,52	60,48
NN	0	66,15	0	62,35	36,52	63,48
SVC	0	80,82	0	65,37	47,72	52,28
DTC	1,66	60,75	21,17	61,95	32,61	67,39
RFC	0	60,75	0	61,56	31,76	68,24
LR	0	82,73	0	65,77	49,01	50,99
Sentimientos	Recall 0	Recall 1	Sensibilidad	Sensibilidad	Precisión	Error
KNN	9,67	74,35	7,14	67,02	45,93	54,07
NN	5,37	77,85	5,74	66,7	47,86	52,14
SVC	1,07	93,8	4,76	67	61,6	38,4
DTC	0	63,11	0	62,56	33,9	66,1
RFC	0	63,11	0	62,56	33,9	66,1
LR	0	93,56	0	66,81	61,12	38,88

4.2.1.2.1 Conclusión:

En la p2 sigue apareciendo el mismo problema que en la p1. El recall de la clase 0 es muy inferior en comparación al de la clase 1, aunque se observa cierta mejoría con respecto a la p1.

Por ello, no se aceptó ningún clasificador como válido.

4.2.1.3 P3

En la Tabla 6 se muestran los resultados de la p3, donde se balanceó el *dataset* añadiendo más elementos de la clase 0.

Tabla 6: PV1 Resultado Prueba 3

TFIDF-100	Recall 0	Recall 1	Sensibilidad	Sensibilidad	Precisión	Error
KNN	46,13	43,73	56,35	46,53	45,2	54,8
NN	17,49	66,03	44,67	45,98	38,1	61,9
SVC	16,45	68,58	56,55	44,53	32,76	67,24
DTC	3,68	75,72	52,09	43,32	29,06	70,94
RFC	3	75,72	50,5	43,22	28,52	71,48
LR	37,57	53,91	57,48	47,75	40,22	59,78
Doc2Vec	Recall 0	Recall 1	Sensibilidad	Sensibilidad	Precisión	Error
KNN	28,92	69,87	41,33	69,29	56,44	43,56
NN	0,67	91,06	20	65,77	57,51	42,48
SVC	15,64	86,77	51,2	69,42	61,34	38,66
DTC	0,66	86,77	21,66	65,28	54,36	45,64
RFC	0,66	86,77	21,66	65,28	54,36	45,64
LR	15,85	88,14	52,97	69,75	62,33	37,67
Sentimiento	Recall 0	Recall 1	Sensibilidad	Sensibilidad	Precisión	Error
KNN	61,98	49,24	61,8	45,03	36,52	63,48
NN	58,77	56,08	65,4	13,94	32,9	67,1
SVC	42,56	65,4	35,21	12,1	20,29	79,71
DTC	20,57	78,36	69,59	40,08	23,49	76,52
RFC	19,88	78,36	36,26	39,99	22,9	77,1
LR	48,02	59,11	32,07	12,37	23,04	76,96

4.2.1.3.1 Conclusión

Hay un aumento de la precisión y del recall de la clase 0 respecto de las pruebas anteriores. Esto se demuestra por ejemplo en KNN para TFIDF-100 donde para la p1 su recall era del 3.48% y posteriormente es del 46.13%. Esto indica que el desbalanceo está afectando de forma negativa a los resultados. Aun así, la diferencia entre recalls es superior al 30% y en las situaciones donde se cumple esta propiedad, la precisión es inferior al 60%.

Por ello no se aceptó ningún clasificador como válido.

4.2.1.4 P4

En la Tabla 7 se muestran los resultados de la p4, donde se balanceó el *dataset*. En esta situación se aumentó el número de casos de la clase 0. Además, se eliminaron los usuarios que solo habían escrito un comentario.

Tabla 7: PV1 Resultado Prueba 4

TFIDF-100	Recall 0	Recall 1	Sensibilidad	Sensibilidad	Precisión	Error
KNN	3,96	75,89	25,033	71,37	51,36	48,64
NN	0	90,5	0	71,72	61,87	38,12
SVC	0	81,92	0	71,54	53,87	46,13
DTC	0	71,01	0	70,57	43,85	56,15
RFC	0	71,01	0	70,57	43,98	56,02
LR	0,3	88,89	2,86	71,69	60,77	93,23
Doc2Vec	Recall 0	Recall 1	Sensibilidad	Sensibilidad	Precisión	Error
KNN	48,57	47,22	56,66	52,68	51,98	48,02
NN	0,27	76,23	20	46,16	31,77	68,23
SVC	35,19	75,74	65,45	56,17	53,06	46,93
DTC	2,15	76,02	60	46,6	32,62	67,38
RFC	1,25	76,23	40	46,41	32,2	67,8
LR	32,84	72,9	63,22	55,03	49,44	50,56
Sentimiento	Recall 0	Recall 1	Sensibilidad	Sensibilidad	Precisión	Error
KNN	9,84	63,29	49,13	50,2	26,96	73,04
NN	4,98	61,47	19,04	17,13	17,92	82,08
SVC	3,11	61,9	15,87	16,74	17,01	82,99
DTC	0,51	69,46	34,78	47,98	25,89	74,11
RFC	0,19	69,46	33,33	47,91	25,71	74,29
LR	3,1	61,9	15,87	16,74	17,01	82,99

4.2.1.4.1 Conclusión

De las tres técnicas la que se ve más exitosa tras las modificaciones es doc2vec, obteniendo una diferencia de recall del 1,25% y una precisión del 51,99 %. Esto se debe a que doc2vec utiliza un *dataset* a nivel de usuario, y la eliminación de un usuario genera un mayor impacto que la eliminación de un comentario, dado que hay más casos de comentarios que de usuarios.

El clasificador KNN en particular obtuvo una gran mejoría con respecto a p1, p2 y p3 llegando a un recall con una diferencia menor al 30%. No obstante, la precisión es menor que 60%.

En las pruebas sin la eliminación de los abandonadores, KNN de doc2vec ha ofrecido el mejor resultado, siendo, por tanto, el clasificador elegido para este contexto, aunque no sobrepase los requisitos mínimos.

4.2.1.5 P5

En la Tabla 8 se muestran los resultados de la p5, donde se eliminaron los abandonadores del *dataset*.

Tabla 8: PV1 Resultado Prueba 5

TFIDF-100	Recall 0	Recall 1	Sensibilidad	Sensibilidad	Precisión	Error
KNN	68,28	36,23	57,94	45,56	52,2	47,8
NN	75,35	26,08	56,9	34,67	52,68	47,32
SVC	82,01	22,7	57,67	46,28	52,81	47,19
DTC	61,35	34,37	55,6	40,93	48,27	51,23
RFC	70,3	26,26	55,2	37,11	48,63	51,37
LR	80,71	22,02	57,38	45,57	51,83	48,17
Doc2Vec	Recall 0	Recall 1	Sensibilidad	Sensibilidad	Precisión	Error
KNN	52,64	69,63	58,34	65,46	67,58	32,42
NN	43,47	76,47	53,58	62,84	63,23	36,77
SVC	42,87	76,73	55,04	63,39	65,97	34,02
DTC	50,65	59,8	49,83	58,83	58,88	41,12
RFC	45,51	71,55	50,2	64,7	65,22	34,78
LR	43,43	74,91	54,39	61,98	67,14	32,86
Sentimiento	Recall 0	Recall 1	Sensibilidad	Sensibilidad	Precisión	Error
KNN	9,84	63,29	49,13	50,2	26,96	73,04
NN	4,98	61,47	19,04	17,13	17,92	82,08
SVC	3,11	61,9	15,87	16,74	17,01	82,98
DTC	0,51	69,46	34,78	47,98	25,89	74,11
RFC	0,19	69,46	33,33	47,91	25,71	74,29
LR	3,11	61,9	15,87	16,74	17,01	82,98

4.2.1.5.1 Conclusión

Salvo la técnica de sentimientos, el resto de las técnicas han dado lugar a mejores resultados en comparación con p1. De este resultado se puede concluir que los abandonadores estaban introduciendo ruido al *dataset*. Como se comentó anteriormente, el porqué del abandono es desconocido. En el caso del sentimiento, al eliminar los abandonadores se observa una cierta mejoría, pero el problema de bajo nivel de sentimiento sigue apareciendo.

De todos los clasificadores obtenidos en esta prueba el que ofrece un mejor rendimiento es KNN para doc2vec que cumple todos los requisitos.

4.2.1.6 P6

En la Tabla 9 se muestran los resultados de la p6 donde se eliminaron los abandonadores y los usuarios con solo un comentario

Tabla 9: PV1 Resultado Prueba 6

TFIDF-100	Recall 0	Recall 1	Sensibilidad 0	Sensibilidad 1	Precisión	Error
KNN	66,09	58,98	56,11	65,13	61,45	38,55
NN	60,82	53,18	65,94	59,07	61,57	38,43
SVC	77,49	46,14	55,21	65,61	59,87	40,13
DTC	65,23	43,6	48,99	57,13	53,68	46,32
RFC	76,98	41,49	52,97	61,78	57,42	42,58
LR	65,15	52,11	54,15	59,9	57,42	42,58
Doc2Vec	Recall 0	Recall 1	Sensibilidad 0	Sensibilidad 1	Precisión	Error
KNN	78,17	39,77	75,2	47,87	68,37	31,63
NN	76,34	65,93	85,16	56,47	71,77	28,23
SVC	92,5	1,81	67,62	20	63,33	36,66
DTC	79,28	39,3	74,57	47,55	64,85	35,14
RFC	81,86	39,31	76,55	49,1	68,37	31,63
LR	96	0	67,92	0	65	35
Sentimiento	Recall 0	Recall 1	Sensibilidad 0	Sensibilidad 1	Precisión	Error
KNN	76,91	31,65	68,21	45,67	57,2	42,8
NN	100	0	65,82	0	65,82	34,17
SVC	100	0	65,82	0	65,82	34,17
DTC	67,67	44,5	69,57	40,73	58,64	41,36
RFC	75,6	35,2	69,15	43,55	60,06	39,94
LR	100	0	65,82	0	65,82	34,17

4.2.1.6.1 Conclusión

Se observa un desbalanceo al eliminar a los usuarios que solo escribieron un comentario. Ahora la clase 1 es minoritaria. Esto produce un aumento de la precisión general debido a que los clasificadores están optando por elegir la clase 0 en un mayor número de ocasiones. Hay dos excepciones:

En primer lugar, TFIDF-100, cuyo *recall* ha aumentado y se ha reducido la diferencia de este valor entre clases. Esto indica que los atributos seleccionados son tan identificativos que el clasificador es capaz de obtener un patrón pese el desbalanceo.

Seguidamente está NN para Doc2Vec, que cumple los requisitos y cuya precisión es la más elevada en esta prueba.

4.2.1.7 P7

En la Tabla 10 se muestran los resultados de la p7, donde se eliminaron abandonadores y se balanceó aumentando los casos de la clase 1.

Tabla 10: PV1 Resultado Prueba 7

TFIDF-100	Recall 0	Recall 1	Sensibilidad 0	Sensibilidad 1	Precisión	Error
KNN	55,45	51,4	42,49	63,46	51,88	48,12
NN	64,88	40,85	41,57	41,57	48,98	51,02
SVC	59,89	45,97	41,99	63,23	49,95	51,05
DTC	61,69	38	39,75	60,02	45,71	54,28
RFC	61,27	38,61	39,64	60,23	45,63	54,37
LR	50,51	54,49	41,75	62,3	50,6	49,4
Doc2Vec	Recall 0	Recall 1	Sensibilidad 0	Sensibilidad 1	Precisión	Error
KNN	35,36	83,27	38,53	76,03	73,62	26,38
NN	33,45	87,34	40,62	75,96	73,87	26,13
SVC	25,51	94,82	42,76	75,76	76,18	23,82
DTC	42,15	60,03	31,52	71,06	60,6	39,4
RFC	35,08	81,3	38,63	75,19	71,83	28,17
LR	26,25	94,2	49,26	75,53	75,4	24,6
Sentimiento	Recall 0	Recall 1	Sensibilidad 0	Sensibilidad 1	Precisión	Error
KNN	55,71	53,73	49,55	58,9	53,32	46,68
NN	64,27	42,06	49,66	33,66	44,75	55,25
SVC	53,29	53,57	52,36	33,13	44,75	55,25
DTC	66,63	27,55	43,87	49,6	45,09	54,91
RFC	62,47	39,09	46,42	55,68	49,5	50,5
LR	48,6	57,64	53,03	65,97	43,8	56,19

4.2.1.7.1 Conclusión

En la prueba anterior, p7, doc2vec no presentaba un problema de desbalanceo. No obstante, tras el aumento del número de casos de la clase 1 se generó un dataset no balanceado. Esto afecta negativamente, puesto que antes estaba balanceado y al añadirle más elementos se vuelve a caer en el desbalanceo.

Aunque ahora la diferencia de recall del resto de pruebas sea menor del 30%, ninguna de las pruebas tiene una precisión mayor al 60%. Esto indica que el balanceo no es una modificación suficiente para mejorar el aprendizaje de los clasificadores o que aún hay datos que generan ruido.

Por tanto, no se aceptó ningún clasificador como válido.

4.2.1.8 P8

En la Tabla 11 se muestran los resultados de la p8, donde se eliminaron abandonadores y los usuarios con solo 1 comentario. Además de balancearse aumentado el número de elementos de la clase 1.

Tabla 11: PV1 Resultado Prueba 8

TFIDF-100	Recall 0	Recall 1	Sensibilidad 0	Sensibilidad 1	Precisión	Error
KNN	66,09	58,97	56,11	65,12	61,45	38,55
NN	60,82	53,18	65,94	59,07	61,57	38,43
SVC	77,49	46,14	55,21	65,61	59,87	40,13
DTC	66,54	41,32	48,86	56,68	53,36	46,64
RFC	76,98	41,49	52,97	61,78	57,42	42,58
LR	65,15	52,1	54,14	59,89	56,68	43,32
Doc2Vec	Recall 0	Recall 1	Sensibilidad 0	Sensibilidad 1	Precisión	Error
KNN	69,92	39,64	73,85	36,01	61,2	38,8
NN	81,11	59,83	81,48	60,55	73,1	26,9
SVC	76,98	37,78	74,6	46,15	65,17	34,83
DTC	68,32	44,78	74,76	35,64	61,23	39
RFC	82,97	39,5	75,85	46,28	69,76	30,24
LR	80,87	12,07	66,19	28,33	59,83	40,16
Sentimiento	Recall 0	Recall 1	Sensibilidad 0	Sensibilidad 1	Precisión	Error
KNN	58,09	56,79	56,2	57,9	58,06	41,94
NN	66,61	45,79	57,59	33,56	50,18	49,82
SVC	54,26	55,89	62,54	31,94	47,67	52,33
DTC	70,95	39,98	53,2	58,24	54,12	45,88
RFC	68,7	44,42	54,33	59,13	55,56	44,44
LR	52,51	55,8	61,54	31,25	46,6	53,4

4.2.1.8.1 Conclusión:

Se eliminaron abandonadores y los usuarios que solo habían escrito un comentario, además de balancearse los casos, aumentado el número de elementos de la clase 1.

Doc2Vec presenta los mismos problemas que en la p7. Por tanto, la modificación adicional no tuvo un gran efecto en el dataset a nivel de usuario. Para el resto de técnicas, la diferencia de *recalls* ha evolucionado positivamente, existiendo ya clasificadores que superan los requisitos mínimos, como es el caso de NN para doc2vec en p6.

En este caso se selecciona NN para TFIDF-100, que presenta un 61.57% de acierto, como el que ofrece mejores resultados en esta prueba.

Las pruebas desde el punto de vista de las 3 clases por separado (PV2) se organizaron de la misma forma que para PV1, con la salvedad de que solo se realizaron dos modificaciones: eliminar usuarios con un solo comentario y balancear el dataset. Las pruebas realizadas se muestran en la tabla 12.

Tabla 12: Pruebas PV 2

Pruebas	BALANCEAR	COMENTARIOS > 1
p1	✗	✗
p2	✗	✓
p3	✓	✗
p4	✓	✓

La información de la situación de las clases en cada prueba, se puede encontrar en los anexos C y D.

4.2.2 PV2

A continuación, se muestran los resultados de todas las pruebas realizadas en las que se abordó desde el punto de vista donde están las tres clases a la vez (PV2)

4.2.2.1 P1

En la Tabla 13 se muestran los resultados de la p1 donde no se realizó ninguna modificación.

Tabla 13: PV2 Resultado Prueba 1

TFIDF	Recall 0	Recall 1	Recall 2	Sensibilidad 0	Sensibilidad 1	Sensibilidad 2	Precisión	Error
KNN	0	0	0	0,18	0	0	0,19	99,81
NN	0	0	0	0	0	0	0	100
SVC	0	0	0	0,18	0	0	0,19	99,81
DTC	0	0	0	0	0	0	0	100
RFC	0	0	0	0	0	0	0	100
LR	0	0	0	1,41	0	0	1,42	98,58
Doc2Vec	Recall 0	Recall 1	Recall 2	Sensibilidad 0	Sensibilidad 1	Sensibilidad 2	Precisión	Error
KNN	0	0	0	0,19	0	0	0,19	99,79
NN	0	0	0	0	0	0	0	100
SVC	0	0	0	0	0	0	0	100
DTC	0	0	0	0,59	0	0	0,59	99,39
RFC	0	0	0	0,59	0	0	0,59	99,39
LR	0	0	0	0	0	0	0	100
Sentimientos	Recall 0	Recall 1	Recall 2	Sensibilidad 0	Sensibilidad 1	Sensibilidad 2	Precisión	Error
KNN	0	0	0	0	0	0	0	100
NN	5,74	0	0	23,93	0	0	13,12	86,88
SVC	5,12	0	0	29,48	0	0	15,27	84,73
DTC	0	0	0	0	0	0	0	100
RFC	0	0	0	0	0	0	0	100
LR	5,12	0	0	29,48	0	0	15,27	84,73

4.2.2.1.1 Conclusión

En los resultados se observa que los clasificadores no son capaces de aprender de forma correcta los datos. Hay que recalcar que para la clase 0 sí se llega a encontrar un patrón. Solo se dio esta situación con la técnica de sentimiento. Esto es indicativo de la existencia de una correlación entre los comentarios y el sentimiento. Se puede suponer que esto es algo positivo, pero no es suficiente para obtener una precisión mínima.

Por tanto, No se aceptó ningún clasificador como válido.

4.2.2.2 P2

En la Tabla 14 se muestran los resultados de la p2, donde se eliminó los usuarios con solo un comentario.

Tabla 14: PV2 Resultado Prueba 2

TFIDF	Recall 0	Recall 1	Recall 2	Sensibilidad 0	Sensibilidad 1	Sensibilidad 2	Precisión	Error
KNN	0	0	0	0	0	0	0	100
NN	0	0	0	0	0	0	0	100
SVC	0	0	0	0	0	0	0	100
DTC	0	0	0	0	0	0	0	100
RFC	0	0	0	0	0	0	0	100
LR	0	0	0	0	0	0	0,44	99,56
Doc2Vec	Recall 0	Recall 1	Recall 2	Sensibilidad 0	Sensibilidad 1	Sensibilidad 2	Precisión	Error
KNN	0	0	0	0	0	0	0	100
NN	0	0	0	0	0	0	0	100
SVC	0	0	0	0	0	0	0	100
DTC	0	0	0	0	0	0	0	100
RFC	0	0	0	0	0	0	0	100
LR	0	0	0	1,73	0	0	1,73	98,27
Sentimientos	Recall 0	Recall 1	Recall 2	Sensibilidad 0	Sensibilidad 1	Sensibilidad 2	Precisión	Error
KNN	0	0	0	0	0	0	0	100
NN	8,1	0	0	20,47	0	0	12,47	87,53
SVC	4,59	0	0	21,9	0	0	11,9	88,01
DTC	0	0	0	0	0	0	0	100
RFC	0	0	0	0	0	0	0	100
LR	3,84	0	0	22,85	0	0	12,23	87,77

4.2.2.2.1 Conclusión

La modificación solo ha tenido efecto en la técnica de sentimientos. Ha habido una cierta mejora en el *recall*, pero no suficiente para pasar los requisitos mínimos.

Por eso, no se aceptó ningún clasificador como válido.

4.2.2.3 P3

En la Tabla 15 se muestran los resultados de la p3, donde se balanceó el *dataset* duplicando los datos de la clase 0 y clase 1.

Tabla 15: PV2 Resultado Prueba 3

TFIDF	Recall 0	Recall 1	Recall 2	Sensibilidad 0	Sensibilidad 1	Sensibilidad 2	Precisión	Error
KNN	0	0	0	0	0	0	0	100
NN	0	0	0	0	0	0	0	100
SVC	0	0	0	0	0	0	0	100
DTC	0	0	0	0	0	0	0	100
RFC	0	0	0	0	0	0	0	100
LR	0	0	0	0	0	0	0,44	99,56
Doc2Vec	Recall 0	Recall 1	Recall 2	Sensibilidad 0	Sensibilidad 1	Sensibilidad 2	Precisión	Error
KNN	0	0	0	0,66	0	0	0,66	99,34
NN	0	0	0	0	0	0	0	100
SVC	0	0	0	0,66	0	0	0,66	99,34
DTC	0	0	0	0,66	0	0	0,66	99,34
RFC	0	0	0	0,66	0	0	0,66	99,34
LR	0	0	0	1,1	0	0	1,1	98,9
Sentimientos	Recall 0	Recall 1	Recall 2	Sensibilidad 0	Sensibilidad 1	Sensibilidad 2	Precisión	Error
KNN	0	0	0	0	0	0	0	100
NN	0	0	0	0	0	0	0	100
SVC	20	0	0	33,33	0	0	1,01	98,99
DTC	0	0	0	0,14	0	0	0,14	99,86
RFC	0	0	0	0,14	0	0	0,14	99,86
LR	0	0	0	0	0	0	0	100

4.2.2.3.1 Conclusión

En estos resultados continúan observándose los mismos problemas que los obtenidos anteriormente. Llegando a este punto, se piensa que existe una cantidad de ruido muy elevada en los datos, haciendo que los clasificadores no puedan aprender de forma exitosa.

Por ello, no se aceptó ningún clasificador como válido.

4.2.2.4 P4

En la Tabla 16 se muestran los resultados de la p4, donde se han eliminado los usuarios con un comentario y se ha balanceado el *dataset* añadiendo elementos de la clase 0 y clase 1.

Tabla 16: PV2 Resultado Prueba 4

TFIDF	Recall 0	Recall 1	Recall 2	Sensibilidad 0	Sensibilidad 1	Sensibilidad 2	Precisión	Error
KNN	0	0,27	0,27	0	0	0	0,1	99,9
NN	0	0	0	0	0	0	0	100
SVC	0	0	0	0	0	0	0	100
DTC	0	0	0	0	0	0	0	100
RFC	0	0	0	0	0	0	0	100
LR	0	0,34	0,34	0	0	0	0,1	99,9
Doc2Vec	Recall 0	Recall 1	Recall 2	Sensibilidad 0	Sensibilidad 1	Sensibilidad 2	Precisión	Error
KNN	0	0	0	0	0	0	0	100
NN	0	0	0	0,52	0	0	0,52	99,48
SVC	0	0	0	9,82	0	0	9,82	90,18
DTC	0	0	0	0	0	0	0	100
RFC	0	0	0	0	0	0	0	100
LR	0	0	0	5,68	0	0	5,68	94,32
Sentimientos	Recall 0	Recall 1	Recall 2	Sensibilidad 0	Sensibilidad 1	Sensibilidad 2	Precisión	Error
KNN	0	0	0	0	0	0	0	100
NN	27,77	0	0	0	0	0	3,2	96,8
SVC	23,8	0	0	0	0	0	1,6	98,4
DTC	0	0	0	0	0	0	0	100
RFC	0	0	0	0	0	0	0	100
LR	23,8	0	0	0	0	0	1,6	98,4

4.2.2.4.1 Conclusión

Se observa una leve mejoría del recall en el caso de la técnica TFIDF para la clase 1 y la clase 2, pero no es suficiente como para que la precisión pase los requisitos mínimos. Los resultados obtenidos con el resto de técnicas siguen presentando los mismos problemas que en las pruebas anteriores.

Por ello, no se aceptó ningún clasificador como válido.

4.2.3 Conclusión del Análisis

Tras la revisión y reflexión de todas las pruebas realizadas, se llegó a las siguientes conclusiones para PV1:

Los abandonadores no introdujeron valores útiles. Esto se puede observar en la gran mejoría que obtuvieron todas las técnicas en su clasificación tras su eliminación. Su aparición introduce dificultad en el aprendizaje de los autómatas y, por tanto, una reducción de la precisión.

Existe una elevada similitud de datos entre clases. Se llegó a esta conclusión porque se necesitó un balanceo para que los clasificadores mejoraran su precisión y recall. De haber existido una mayor diferencia entre los datos de cada clase, los autómatas habrían sabido diferenciar las agrupaciones pese al desbalanceo.

Se observa un desbalanceo en el dataset. Se puede observar por la alta diferencia de recall de todas las pruebas cuyo estado del dataset estaba desbalanceado. Esto genera que los clasificadores tiendan a predecir la clase mayoritaria.

Bajo nivel de sentimiento en los comentarios. Todas las pruebas usando esta técnica obtuvieron resultados de precisión bajos comparados con el resto. Por tanto, el problema radica en los atributos utilizados, los cuales no tienen la suficiente potencia como para que el autómata distinga los patrones sentimentales en los comentarios.

Doc2vec se revela como la mejor técnica. Con los resultados obtenidos de manera general, siempre se situó por encima del resto. Esto es debido a la potencia inherente del algoritmo y a su menor sensibilidad a problemas de desbalanceo (ya que opera a nivel de usuario).

En todas las modificaciones omitiendo la eliminación de abandonadores, ambas mejoraron el aprendizaje.

A nivel de usuario, la modificación que más destacó fue la eliminación de los usuarios que solo habían escrito un comentario. De forma similar a los abandonadores, introdujeron ruido en el dataset.

A nivel de comentarios, ambas modificaciones juntas funcionaron exitosamente. Además de lo anterior, el dataset estaba más desbalanceado y era necesaria una corrección.

Desde el punto de vista de tres clases en PV2 se concluyó que:

La clase 2 genera incertidumbre al intentar clasificar sus datos. Esto se puede observar en la baja precisión obtenida en todas las pruebas en general, no llegando a superar el nivel mínimo de precisión. Además, haciendo una omisión a los usuarios que abandonan se puede visualizar una mejora de hasta 70% (p6).

Ninguna de las modificaciones que se realizaron en los dataset fueron suficientes para mejorar la precisión y recall de los clasificadores.

Tras todas las reflexiones anteriores, considerando los resultados obtenidos en el conjunto de pruebas realizadas en este trabajo, se puede concluir que la respuesta a la pregunta “¿Se ha podido predecir si los usuarios aprobarán, suspenderán o fracasarán en un MOOC en función de los comentarios que escriben?” es no, al menos utilizando las técnicas y algoritmos presentados en este trabajo y los datos disponibles en los dos MOOCs analizados. El motivo es que los abandonadores generan una distorsión muy elevada en el dataset y solo utilizando KNN con doc2vec en p4 se obtuvieron los mejores resultados, teniendo una diferencia de recall menos al 30% y una precisión mayor al 50%, pero sin pasar todos los requisitos mínimos establecidos.

Sí se ha demostrado que se puede detectar a los usuarios finalizadores, obteniendo un modelo que logra identificarlos y que supera los requisitos mínimos con un 71% de precisión en la **p6**, eliminando los usuarios que solo comentaron una vez.

5 CONCLUSIÓN

A raíz del análisis anterior y considerando las conclusiones presentadas en el apartado anterior (sección 4.2.3), se han obtenido una serie de conclusiones generales que se espera puedan influir de forma positiva en posteriores análisis con idéntica o similar temática.

Se descubrió que el nivel de sentimiento interno plasmado en los comentarios de los estudiantes era muy bajo, algo comprensible teniendo en cuenta que el contexto de los comentarios era académico. Si se estuviera recogiendo la información de una red social informal, como Facebook o Twitter, el nivel de sentimiento interno que quedaría reflejado en los textos sería mucho más elevado. También podría deberse a que la librería seleccionada para el análisis de sentimientos en este trabajo esté ofreciendo poca sensibilidad al tipo de palabras contenidas en los comentarios de los estudiantes en el MOOC. Para resolver esto, se podría crear un diccionario de sentimientos y generar un algoritmo propio. Para ello sería necesario realizar una investigación sobre el nivel de sentimiento de las palabras y sobre algoritmos para analizar sinápticamente frases.

Otra posibilidad sería que el bajo acierto de los clasificadores también se deba a falta de información. Esto generaría que ciertos grupos de datos tuvieran una baja representación dentro del *dataset*, pudiendo necesitar más casos para poder reconocerlos y aprender de ellos, además de resolverse el problema del desbalanceo.

Aunque para el contexto donde no había abandonadores se han obtenido buenos resultados, podría haber problema con la configuración de los clasificadores. Como se ha explicado anteriormente, éstos han de estar debidamente configurados y, de no ser así, no son lo suficientemente precisos.

Se podrían haber introducido datos artificiales: nueva información creada por el investigador para solventar la incertidumbre, el desbalanceo y la falta de sentimientos. En mi opinión, esta solución distorsionaría la realidad aún más que las modificaciones que se realizaron.

Otro posible cambio sería realizar una configuración más avanzada del dispositivo electrónico con el que se programó. *Bag of words* se tuvo que omitir por problemas de almacenamiento. Si se configurase la administración de la memoria del sistema operativo se podría haber llegado a usar esta técnica. Aun así, este método de NLP hubiera sido altamente costoso de procesamiento para el aprendizaje de todos los clasificadores.

No se hizo un filtrado de los comentarios y usuarios que han abandonado. Si se hiciera, se estarían omitiendo todos aquellos datos cuya información era inválida o confusa. Esto ayudaría a los clasificadores a realizar un aprendizaje efectivo, eliminando el ruido del *dataset*. En mi opinión, esta solución también alteraría la realidad aún más que las modificaciones realizadas, además de requerir una gran cantidad de tiempo.

Hay que remarcar que, de toda la información ofrecida por los MOOCs, solo se usaron los comentarios, sin tener en cuenta su contexto. Existen otros trabajos que sí utilizan la potencia de los datos de los cursos y, en mi opinión, se podrían complementar con los clasificadores encontrados.

Para la traducción de los textos a vectores se emplearon las técnicas de NLP anteriormente descritas. Pero se podrían haber creado expresiones personalizadas por el investigador en referencia a un análisis manual de toda la información ofrecida por los cursos online y el contexto de los comentarios. Así se orientaría el vector al contexto del tema que se estuviera tratando en las clases virtuales. Sin embargo, esta técnica perdería potencia tras un cambio de entorno académico.

A modo de resumen, las líneas posibles de trabajo a explorar en el futuro serían:

- Buscar más información para que las clases tengan más representación dentro del *dataset*.
- Generar, si fuera necesario, nuevos casos artificiales para balancear el *dataset*.

Existen otras posibilidades que, aunque creo que presentarían las desventajas o dificultades mencionadas anteriormente, también se podrían explorar para comprobar si se lograría mejorar los resultados o no. Estas son:

- Crear una librería propia para el análisis de sentimiento.
- Realizar un filtrado sobre los datos de los abandonadores para eliminar ruido.
- Buscar un vector personalizado, basándose en técnicas de NLP sobre los textos.

Finalmente, para concluir, este trabajo me ha permitido profundizar en los conocimientos sobre aprendizaje automático aprendidos durante mis estudios en la universidad y aprender técnicas nuevas. Además, he podido experimentar, a un nivel más humilde, cómo un investigador afronta sus retos y trata de resolver los problemas que encuentra durante su investigación. He seguido todos los pasos necesarios para el desarrollo de un proyecto, desde la idea inicial hasta su finalización.

He aprendido que un resultado en el que la conclusión es que no se puede hacer lo que se pretendía inicialmente, también es un resultado válido. Aun así, he sentido mucha frustración a la hora de buscar atributos para el aprendizaje automático y probar distintos algoritmos y configuraciones, pues con ninguno de ellos lograba obtener los resultados deseados, quedándome cierta sensación de fracaso...

REFERENCIAS

Alias, Gerardo.,Cassanelli, Rodrigo (2019). *NLP aplicado a análisis de texto. Proyecto Final de Universidad Nacional de Mar del Plata*. Facultad de Ingeniería. Recuperado de <http://rinfi.fi.mdp.edu.ar/xmlui/handle/123456789/354>

Baentos Martínez, Rocío Erandi., Cruz Ramírez, Nicandro., Acosta Mesa, Héctor Gabriel., Rabatte Suárez, Ivonne., Gogeochea Trejo, M^a del Carmen., Pavón León, Patricia., Blázquez Morales, Sobeida. (2009). Árboles de decisión como herramienta en el diagnóstico médico. En *Artículo Original* 20-24

Baviera, Tomás (2017). Técnicas para el análisis del sentimiento en Twitter: Aprendizaje Automático Supervisado y SentiStrength. en *DIGITOS. Revista de Comunicación Digital* 3 33-50.

Beyer, Kevin., Goldstein, Jonathan., Ramakrishnan, Raghu., Shaft, Uri (1998). *When is Nearest neighbor Meaningful?*. Beerli C., Buneman P. (eds) Database Theory Cynthia Lorena, Corso (2017). Aplicación de algoritmos de clasificación supervisada usando Weka. Universidad Tecnológica Nacional, Facultad Regional Córdoba. Recuperado de: https://www.investigacion.frc.utn.edu.ar/labsis/Publicaciones/congresos_labsis/cynthia/CN_IT_2009_Aplicacion_Algoritmos_Weka.pdf

García Cambronero, Cristina., Gómez Moreno, Irene (2016). Algoritmos de Aprendizaje: KNN & KMeans Recuperado de <http://www.it.uc3m.es/jvillena/irc/practicas/08-09/06.pdf>

Han Lau, Jey ., Baldwin, Timothy. (2016) *An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation* . Dept of Computing and Information Systems, The University of Melbourne.

Hanna M. Wallach (2006). *Topic Modeling: Beyond Bag-of-Words*. En University of Cambridge

Isidro Estradas, Cristina (2017). *Propuesta de un método basado en Deep Learning para Learning Analytics en MOOCs (Trabajo Final de Máster)*. Universidad Autónoma de Madrid.

Jordan, Katy (2015). Massive Open Online Course Completion Rates Revisited: Assessment, Length and Attrition. En *International Review of Research in Open and Distributed Learning* Volume 16, Number 3,

Jurafsky, Daniel., Martin, James H., (2019). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Londres. Pearson International.

Loria, Steven (2020). Textblob Documentation. Recuperado de: <https://buildmedia.readthedocs.org/media/pdf/textblob/latest/textblob.pdf>

- Maas, Andrew L., Daly, Raymond E., Pham, Peter T., Huang, Dan., Y. Ng, Andrew., Potts, Christopher (2011). Learning Word Vectors for Sentiment Analysis. *Association for Computational Linguistics*. 142-150
- Moreno, Antonio., Armengol, Eva., Béjar Javier., Belanche, Luís., Cortés, Ulises., Gavalda, Ricard., Gimeno, Juan Manuel., López, Beatriz., Martín, Mario., Sánchez, Miquel (1994): *Aprendizaje automático*. Politext. Barcelona
- Ortigosa, Álvaro., Carro, Rosa., Bravo-Agapito, Javier., Lizcano, David., Alcolea, Juan Jesús., Blanco, Óscar (2019). From Lab to Production: Lessons Learnt and Real Life Challenges of an Early Student-Dropout Prevention System. *IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES*, 12(2) 264- 277
- Ortigosa, Álvaro., Martín, Jose M., Carro, Rosa M. (2014): Sentiment analysis in Facebook and its application to e-learning. *Computers in Human Behavior* 31. 527-541
- Pedregosa, Fabian., Varoquaux, Gaël., Gramfort Alexandre, Michel Vincent., Thirion Bertrand (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 2825-283
- Saif, Hassan., He, Yulan., Alani, Harith (2012). Semantic Sentiment Analysis of Twitter. En *The Semantic Web--ISWC 2012*. 7649. 508-524. 10.1007/978-3-642-35176-1_32.
- Salas, Rodrigo (2004). Redes Neuronales Artificiales. *Departamento de Computación*. Recuperado de: https://www.academia.edu/24633757/Redes_Neuronales_Artificiales
- Sánchez Solís, Kevin David., Carabalí, Juan Pablo., Jaramillo Valbuena, Sonia., (2018). Análisis de sentimientos haciendo uso de MapReduce. *ESPACIOS*. 39 (45) 35-43
- Soucy, Pascal., Mineau, Guy M. Beyond (2015) TFIDF Weighting for Text Categorization in the Vector Space Model. En *academia.edu*.
- Wainberg, Michael., Alipanahi, Babak., Frey, Brendan J. (2016) Are Random Forests Truly the Best Classifiers. *Journal of Machine Learning Research* 17 1-5.
- Zayat Mata, Ana (2019). *Análisis Automático de Experiencia de Cliente en los Canales de Atención Digitales (Trabajo de Fin de Máster)*. Universidad Autónoma de Madrid

GLOSARIO

Sensible	Nivel de aceptación de cambios de un modelo de clasificación.
Robusto	Capacidad para no verse alterado por cambios.
Dataset	Conjunto de datos.
Precisión	Porcentaje de acierto.
Recall	Porcentaje de casos de una clase en particular que se han clasificado correctamente con respecto del total de esa clase.
Sensibilidad	Que casos respecto de una clase en particular se han acertado respecto de la cantidad de datos que se han clasificado como dicha clase.
Matriz de confusión	Nos indica los resultados del aprendizaje , mostrando sobre qué clases se ha obtenido una clasificación correcta y cuáles se han confundido con otra clase.
Ruido	Información corrupta o que distorsiona los datos dentro del <i>dataset</i> .

ANEXOS

Anexo A – Dataset Pv1 Comentarios

P1

Clases	Nº Comentarios
0	488
1	1215

P2

Clases	Nº Comentarios
0	452
1	975

P3

Clases	Nº Comentarios
0	1464
1	1215

P4

Clases	Nº Comentarios
0	904
1	975

P5

Clases	Nº Comentarios
0	488
1	365

P6

Clases	Nº Comentarios
0	452
1	265

P7

Clases	Nº Comentarios
0	452
1	529

P8

Clases	Nº Comentarios
0	452
1	529

Anexo B – Dataset Pv1 Usuarios

P1

Clases	Nº Usuarios
0	116
1	394

P2

Clases	Nº Usuarios
0	80
1	154

P3

Clases	Nº Usuarios
0	348
1	394

P4

Clases	Nº Usuarios
0	160
1	154

P5

Clases	Nº Usuarios
0	116
1	138

P6

Clases	Nº Usuarios
0	80
1	38

P7

Clases	Nº Usuarios
0	160
1	38

P8

Clases	Nº Usuarios
0	80
1	76

Anexo C – Dataset Pv2 Comentarios

P1

Clases	Nº Comentarios
0	461
1	353
2	811

P2

Clases	Nº Comentarios
0	429
1	256
2	678

P3

Clases	Nº Comentarios
0	922
1	705
2	811

P4

Clases	Nº Comentarios
0	858
1	511
2	678

Anexo D – Dataset Pv2 Usuarios

P1

Clases	Nº Usuarios
0	116
1	139
2	255

P2

Clases	Nº Usuarios
0	80
1	39
2	115

P3

Clases	Nº Usuarios
0	116
1	277
2	511

P4

Clases	Nº Usuarios
0	80
1	77
2	231